IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: **Stephen Fife Sheldon**	§	Group Art Unit: **2168**
	§	
Serial No.: **10/800,493**	§	
	§	Examiner: **Sanders, Aaron J.**
Filing Date: **March 15, 2004**	§	
	§	
Title: **Simple Expression Modification in a SQL Query**	§	Attorney. Docket No.: **11466**
	§	

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPELLANT'S APPEAL BRIEF (37 C.F.R. § 41.37)

This brief is in support of Appellant's notice of appeal from the Final Office Action mailed April 28, 2008.

REAL PARTY IN INTEREST

The real party in interest is:

Teradata US, Inc
1700 S. Patterson Blvd
Dayton, Ohio 45479

RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences, to Appellants' knowledge.

STATUS OF CLAIMS

The application as originally filed contained 42 claims. Claim 2-6, 16-20, and 30-34 were previously canceled. Claims 1, 7-15, 21-29, and 35-42 are pending. Claims 1, 7-15, 21-29, and 35-42 are appealed.

STATUS OF AMENDMENTS

Appellant's Response to Final Office Action filed on July 28, 2008 contained

amendments. Those amendments, which were to correct typographical errors and which were suggested in the Final Office Action, were not entered.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Claim 1 features a method of processing a database query. *Passim*; page 5, lines 15-29; Fig. 3. The query includes an expression. Page 6, lines 2-3; page 6, line 29 – page 7, line 15; Fig. 5, elements 515 and 555. The method includes performing expression optimization on the expression. Page 6, line 1 – page 15, line 27; Figs. 4A, 4B, 5-10, 11A, 11B, 12 and 13. The method further includes performing further query optimization to produce a result. Page 5, lines 21-29; page 17, lines 3-8; page 19, lines 5-10; page 21, lines 9-14; Fig. 3, element 325. The method further includes saving the result in a memory. Page 4, line 20 – page 5, lines 10; knowledge of person of ordinary skill. The method further includes performing expression optimization before further query optimization. Page 5, lines 19-29; Fig. 3, elements 320 and 325. The expression includes a sub-expressions ("SE"). Page 6, lines 2-3; page 14, lines 23-24, cancelled claims 2, 16, and 20. Expression optimization includes representing the query as a tree structure. Page 6, lines 12-28; Figs. 4A and 4B. Expression optimization further includes representing the expression in the tree structure as a parent node having a first child node and a second child node, where the first child node represents the sub-expression, the second child node represents the portion of the expression that is not the sub-expression, and the parent node represents an operation between the first child node and the second child node. Page 6, lines 12-28; Figs. 4A and 4B. Expression optimization further includes determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction and in response, removing the parent node and its children from the tree structure, and inserting the first child node in its place. Page 12, lines 18-25; page 13, line 24 – page 14, line 5; Fig. 11B, elements 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146.

Claim 15 features a computer program, stored on a tangible storage medium, for use in processing a database query. *Passim*; page 5, lines 15-29; Fig. 3. The query includes an expression. Page 6, lines 2-3; page 6, line 29 – page 7, line 15; Fig. 5, elements 515 and 555. The computer program includes executable instructions that cause a computer to perform expression optimization on the expressions. Page 6, line 1 – page 15, line 27; Figs. 4A, 4B, 5-

10, 11A, 11B, 12 and 13. The computer program includes executable instructions that cause a computer to perform further query optimization to produce a result. Page 5, lines 21-29; page 17, lines 3-8; page 19, lines 5-10; page 21, lines 9-14; Fig. 3, element 325. The computer program includes executable instructions that cause a computer to save the result in a memory. Page 4, line 20 – page 5, lines 10; knowledge of person of ordinary skill. The expression includes a sub-expression ("SE"). Page 6, lines 2-3; page 14, lines 23-24, cancelled claims 2, 16, and 20. Expression optimization is performed before further query optimization. Page 5, lines 19-29; Fig. 3, elements 320 and 325. The computer program includes executable instructions that cause a computer to represent the query as a tree structure. Page 6, lines 12-28; Figs. 4A and 4B. The computer program includes executable instructions that cause the computer to represent the expression in the tree structure as a parent node having a first child node and a second child node, where the first child node represents the sub-expression, the second child node represents the portion of the expression that is not the sub-expression, and the parent node represents an operation between the first child node and the second child node. Page 6, lines 12-28; Figs. 4A and 4B. The computer program includes executable instructions that cause the computer to determine that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction and in response, remove the parent node and its children from the tree structure and insert the first child node in its place. Page 12, lines 18-25; page 13, line 24 – page 14, line 5; Fig. 11B, elements 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146.

Claim 59 features a database system including a massively parallel processing system. Page 4, lines 1-6; Fig. 1. The massively parallel processing system includes one or more nodes. Page 4, lines 6-7 and 22-23; Fig. 1, elements $105_{1...O}$. The massively parallel processing system further includes a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs. Page 4, lines 9-19; Fig. 1, elements $110_{1...N}$. The massively parallel processing system further includes a plurality of data storage facilities each of the one or more CPUs providing access to one or more data storage facilities. Page 4, lines 24-26; Fig. 1, elements $120_{1...N}$. The massively parallel processing system further includes a process for execution on the massively parallel processing system for processing a database query, the query including an expressions. Page 6, lines 2-3; page 6, line 29 – page 7, line 15; Fig. 5, elements 515 and 555. The process including performing expression optimization on the expressions. Page 6, line 1 –

page 15, line 27; Figs. 4A, 4B, 5-10, 11A, 11B, 12 and 13. The process including performing further query optimization to produce a result. Page 5, lines 21-29; page 17, lines 3-8; page 19, lines 5-10; page 21, lines 9-14; Fig. 3, element 325. The process including saving the result in a memory. Page 4, line 20 – page 5, lines 10; knowledge of person of ordinary skill. Expression optimization is performed before the further query optimization. Page 5, lines 19-29; Fig. 3, elements 320 and 325. The expression includes a sub-expression ("SE"). Page 6, lines 2-3; page 14, lines 23-24, cancelled claims 2, 16, and 20. Expression optimization includes representing the query as a tree structure. Page 6, lines 12-28; Figs. 4A and 4B. Expression optimization includes representing the expression in the tree structure as a parent node having a first child node and a second child node, where the first child node represents the sub-expression, the second child node represents the portion of the expression that is not the sub-expression, and the parent node represents an operation between the first child node and the second child node. Page 6, lines 12-28; Figs. 4A and 4B. Expression optimization further includes determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction and, in response, removing the parent node and its children from the tree structure and inserting the first child node in its place. Page 12, lines 18-25; page 13, line 24 – page 14, line 5; Fig. 11B, elements 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 1, 15, and 29 are enabled under 35 USC 112, first paragraph.

2. Whether claims 1, 7-15, 21-29, and 35-42 are statutory subject matter under 35 USC 101.

3. Whether claims 1, 7-9, 11-13, 15, 21-23, 25-27, 29, 35-37, and 39-41 are obvious under 35 USC 103(a) over Paulley in view of Warner.

4. Whether claims 10, 14, 24, 28, 38, and 42 are obvious under 35 USC 103(a) over Paulley in view of Nuutila.

5. Whether the drawings are adequate.

ARGUMENT

I. Claims 1, 15, and 29 are enabled by the specification and by what would be known to one of ordinary skill in the art.

The Final Office Action rejected claims 1, 15, and 29 under 35 USC 112, first paragraph, as failing to comply with the enablement requirement, arguing that "the limitations 'performing further query optimization to produce a result' and 'saving the result in a memory' do not appear in the specification." Final Office Action at 3 (emphasis in original).

With respect to the "performing further query optimization to produce a result" limitation, one example of a "result" is shown in Fig. 3 as the output of element 325 ("executable steps"). July 28, 2008 Response at 8. The Advisory Action disagreed, arguing that "this is not clear because it is not clear that the 'optimization' in para. 20 corresponds to the 'further query optimization.'" Advisory Action at 2.

The fact that "further query optimization" corresponds to the "optimization" in paragraph [0020] is illustrated by referring to the patent application as originally filed. Claim 10 states that "further query optimization" includes "determining a transitive closure of the database query." In paragraph [0020] lines 24-25, the specification states that "the optimizer (block 325) determines the transitive closure of the SQL query." Thus, for at least this reason, "further query optimization" corresponds to the "optimization" in paragraph [0020]. Claims 9 and 11 and paragraph [0020] lines 21-29 provide further examples of "further query optimization."

The "saving the result in a memory" limitation is enabled because a person of ordinary skill would understand that the "executable steps" may be saved in a memory so that they can be acted on by the dispatcher, element 210 shown in Fig. 2. July 28, 2008 Response at 8. The Advisory Action disagreed arguing that "[j]ust because 'a person of ordinary skill would understand that the "executable steps" may be saved in a memory' . . . does not make the limitation part of the Appellant's invention." Advisory Action at 2.

The Advisory Action's argument is contrary to Federal Circuit law, which states that the scope of enablement is not limited to what is disclosed in the specification; it also includes "what would be known to one of ordinary skill in the art without undue experimentation." *Abbott Labs v. Andrx Pharmaceuticals, Inc.*, 452 F.3d 1331, 1341 (Fed. Cir. 2006), *citing Invitrogen Corp. v. Clontech Labs., Inc.*, 429 F.3d 1052, 1070-71 (Fed. Cir. 2005). The Advisory Action does not

argue with Appellant's contention that a person of ordinary skill would understand that the "executable steps" may be saved in a memory so that they can be acted on by the dispatcher, element 210 shown in Fig. 2. For at least this reason, the "saving the result in a memory" element of is enabled.

Appellant respectfully requests that this rejection be reversed.

II. The pending claims are patentable subject matter under 35 USC 101.

The Final Office Action rejected all pending claims under 35 USC 101 arguing that the claimed invention is directed to non-statutory subject matter because the disclosed subject matter fails to produce a tangible result. Final Office Action at 4. This rejection appears to arise from the previous rejection, i.e., that the "produce a result" element is not enabled. The enablement rejection should be reversed, as discussed above. Accordingly, the rejection under 35 USC 101 should be reversed as well.

Further, the Final Office Action appears to have applied the "useful, concrete and tangible result" test for patentable subject matter that was recently rejected by the United States Circuit Court for the Federal Circuit. *See In re Bilski*, No. 2007-1130, slip op. at 20 (Fed. Cir. October 30, 2008)("'useful, concrete and tangible result' inquiry is inadequate"). This rejection should be reversed on that basis as well.

The Final Office Action made further arguments regarding the rejection under 35 USC 101, which appear to duplicate arguments made earlier in the Final Office Action. Office Action at 23.

III. Claims 1, 7-9, 11-13, 15, 21-23, 25-27, 29, 35-37, and 39-41 are not obvious over Paulley in view of Warner because "propagating" is not the same as "removing and inserting"

The Final Office Action rejected claims 1, 7-9, 11-13, 15, 21-23, 25-27, 29, 35-37, and 39-41 under 35 USC 103(a) as being unpatentable over United States Patent No. 6,665,664 ("hereinafter Paulley") in view of United States Patent Publication No. 2005/0055338 (hereinafter "Warner"). Final Office Action at 5.

The Final Office Action's combination of Paulley and Warner does not teach or suggest determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction, and in response, removing the parent node and its children from the tree structure and inserting

the first child node in its place, as required by independent claims 1, 15, and 29. The Final Office Action admits that Paulley does not teach this element. Final Office Action at 7. Warner does not teach it either. Even assuming the value of one of the nodes 108 or 110 in Warner's Fig. 1 is "0," Warner does not teach "removing the parent node and its children from the tree structure and inserting the first child node in its place," as required by independent claims 1, 15, and 29. Warner teaches "propagating" the result of the evaluation of the operator at 104 up the tree. Warner at paragraph [0020], lines 10-13. "Propagating" is not the same as "removing . . . and inserting." In particular, Warner does not teach removing nodes 104, 108, and 110 and inserting one of the child nodes (108 or 110) in its place.

The Advisory Action admitted that "Warner and Paulley may not explicitly teach 'that the second child node represents the constant 0,'" but argued that "it is obvious that the referenced 'B' could have the value '0,'" and that in that situation, "the value 'A' at 108 would replace the value '+' at 104, see Warner para. 6." The Advisory Action argued that "[a]n example of this process is depicted in Fig. 3C." Advisory Action at 2. The Advisory Action's argument is not materially different from that made in the Final Office Action and does not meet the substance of Appellant's argument. The cited material from Warner describes propagating, not removing and inserting, as required by the rejected claims.

For at least this reason, the Office Action's combination of Paulley and Warner does not render claims 1, 15, and 29 obvious. Claims 7-9, 11-13, 21-23, 25-27, 35-37, and 39-41 depend from one of claims 1, 15, and 29 and are patentable for at least the same reason. Appellant respectfully requests that this rejection be reversed.

IV. Claims 10, 14, 24, 28, 38, and 42 are not obvious over Paulley in view of Nuutila for the same reason.

The Final Office Action rejected claims 10, 14, 24, 28, 38, and 42 under 35 USC 103(a) as being unpatentable over Paulley in view of Esko Nuutila, "Transitive Closure," Helsinki University of Technology, (October 9, 1995)(hereinafter "Nuutila"). Final Office Action at 18.

Claims 10, 14, 24, 28, 38, and 42 depend from one of claims 1, 15, and 29. The Office Action admits that Paulley is missing several elements from claims 1, 15, and 29, *see* Office Action at 6-8, 10-12, and 15-17, and does not suggest that Nuutila provides the missing elements. In its July 28, 2008 Response, Appellant noted its assumption that the Office Action intended to reject claims 10, 14, 24, 28, 38, and 42 under 35 USC 103(a) as being unpatentable

over Paulley in view of Warner and further in view of Nuutila. The Advisory Action did not correct the assumption.

Appellant respectfully disagrees. As described above, the Office Action's combination of Paulley and Warner does not teach at least one element of claims 1, 15, and 29. The Office Action does not argue that Nuutila provides the missing element. Accordingly, claims 10, 14, 24, 28, 38, and 42 are patentable for at least the reasons described above for claims 1, 15, and 29. Appellant respectfully requests that this rejection be reversed.

VI. The drawings are sufficient

The Final Office Action argued that the drawings do not show the method of the independent claims. The Final Office Action argued that "[r]eferenced Fig. 1 does not show a method, and does not contain elements 320 and 325." Final Office Action at 22. In its July 28, 2008 Response, Appellant disagreed, arguing that:

- Fig. 1 shows the Parsing Engine 130, which is shown in greater detail in Fig. 2;
- Fig. 2 shows a Parser 205, which is shown in greater detail in Fig. 3; and
- Fig. 3 includes elements 320 and 325.

The Advisory Action did not answer this argument. Presumably, the objection to the drawings has been withdrawn.

The Final Office Action further argued that "it is unclear which steps of Figs. 5-6 and 11A-12 correspond to at least the claim limitations 'performing expression optimization,' 'performing further query optimization,' 'saving the result,' 'representing the query as a tree structure,' and 'representing the expression . . . as a parent node having . . . child node[s].'" Final Office Action at 22. In its July 28, 2008 Response, Appellant responded that "flow charts or source code listings are not a requirement for adequately disclosing the functions of software." *Fonar Corp. v. General Electric Co.*, 107 F.3d 1543, 1549 (Fed. Cir. 1997) (best mode context); MPEP 2163(I)(A). Embodiments of claim limitations "performing expression optimization" and "performing further query optimization" are shown in Fig. 3, elements 320 and 325 and in Figs. 5-13. An embodiment of the element "saving the result" is shown as the output of element 325, as discussed above. Embodiments of the elements "representing the query as a tree structure" and "representing the expression in the tree structure as a parent node having a first child node and a second child node" are shown in Figs. 4A and 4B.

The Advisory Action did not respond to this argument. Presumably, this objection has been withdrawn. If not, however, Appellant requests that this objection be reversed.

Summary

In light of the foregoing, Appellant respectfully requests that the final rejection of the pending claims be reversed and the application be remanded for allowance of the pending claims, or, alternatively, that the application be remanded for further examination if appropriate references can be found by the examiner.

Appellant requests that the fees necessary with this filing be debited from deposit account number 50-4370.

Respectfully submitted,

/Howard L. Speight/ Reg. No. 37,733
Howard L. Speight
Reg. No. 37,733
9601 Katy Freeway
Suite 280
Houston, Texas 77024
Telephone: (713) 881-9600
Facsimile: (713) 715-7384
E.Mail: howard@hspeight.com
ATTORNEY FOR APPELLANT

Date: November 7, 2008

CLAIMS APPENDIX

1. A method of processing a database query, the query including an expression, the method including:

performing expression optimization the expression;

performing further query optimization to produce a result;

saving the result in a memory;

performing expression optimization before further query optimization; and

where the expression includes a sub-expressions ("SE"), and where the expression optimization includes:

representing the query as a tree structure;

representing the expression in the tree structure as a parent node having a first child node and a second child node;

where the first child node represents the sub-expression;

where the second child node represents the portion of the expression that is not the sub-expression; and

where the parent node represents an operation between the first child node and the second child node;

determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction; and

in response, removing the parent node and its children from the tree structure and inserting the first child node in its place.

7. The method of claim 1, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause.

8. The method of claim 1, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause.

9. The method of claim 1, where further query optimization includes:

determining a satisfiability of the database query.

10. The method of claim 1, where further query optimization includes:
determining a transitive closure of the database query.

11. The method of claim 1, where further query optimization includes:
determining one or more plans for executing the query.

12. The method of claim 11, where one of the one or more plans includes:
scanning a table to locate rows that satisfy one or more conditions; and
summing one or more columns in the rows that satisfy the one or more conditions.

13. The method of claim 1, where further query optimization includes:
selecting an optimal plan from executing the database query.

14. The method of claim 1, where further query optimization includes two or more optimizations selected from the group consisting of:
determining a satisfiability of the database query;
determining a transitive closure of the database query;
determining one or more plans for executing the query; and
selecting an optimal plan from executing the database query.

15. A computer program, stored on a tangible storage medium, for use in processing a database query, the query including an expression, the computer program including executable instructions that cause a computer to:

perform expression optimization on the expressions;

perform further query optimization to produce a result;

save the result in a memory;

where the expression includes a sub-expression ("SE"), where expression optimization is performed before further query optimization, and where the computer program includes executable instructions that cause a computer to:

represent the query as a tree structure;

represent the expression in the tree structure as a parent node having a first child node and a second child node;

where the first child node represents the sub-expression;

where the second child node represents the portion of the expression that is not the sub-expression; and

where the parent node represents an operation between the first child node and the second child node;

determine that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction; and

in response, remove the parent node and its children from the tree structure and insert the first child node in its place.

21. The computer program of claim 15, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause.

22. The computer program of claim 15, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause.

23. The computer program of claim 15, where further query optimization includes:

determining a satisfiability of the database query.

24. The computer program of claim 15, where further query optimization includes:
determining a transitive closure of the database query.

25. The computer program of claim 15, where further query optimization includes:
determining one or more plans for executing the query.

26. The computer program of claim 25, where one of the one or more plans includes:
scanning a table to locate rows that satisfy one or more conditions; and
summing one or more columns in the rows that satisfy the one or more conditions.

27. The computer program of claim 15, where further query optimization includes:
selecting an optimal plan from executing the database query.

28. The computer program of claim 15, where further query optimization includes two or more optimizations selected from the group consisting of:
determining a satisfiability of the database query;
determining a transitive closure of the database query;
determining one or more plans for executing the query; and
selecting an optimal plan from executing the database query.

29. A database system including:
a massively parallel processing system including:
one or more nodes;
a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;
a plurality of data storage facilities each of the one or more CPUs providing access to one or more data storage facilities;
a process for execution on the massively parallel processing system for processing a database query, the query including an expressions, the process including:
performing expression optimization on the expressions;
performing further query optimization to produce a result;
saving the result in a memory;

where the expression optimization is performed before the further query optimization; and

where the expression includes a sub-expression ("SE"), and where expression optimization includes:

representing the query as a tree structure;

representing the expression in the tree structure as a parent node having a first child node and a second child node;

where the first child node represents the sub-expression;

where the second child node represents the portion of the expression that is not the sub-expression; and

where the parent node represents an operation between the first child node and the second child node;

determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction; and

in response, removing the parent node and its children from the tree structure and inserting the first child node in its place.

35. The database system of claim 29, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause.

36. The database system of claim 29, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause.

37. The database system of claim 29, where further query optimization includes:
determining a satisfiability of the database query.

38. The database system of claim 29, where further query optimization includes:
determining a transitive closure of the database query.

39. The database system of claim 29, where further query optimization includes:
determining one or more plans for executing the query.

40. The database system of claim 39, where one of the one or more plans includes:

scanning a table to locate rows that satisfy one or more conditions; and

summing one or more columns in the rows that satisfy the one or more conditions.

41. The database system of claim 29, where further query optimization includes:

selecting an optimal plan from executing the database query.

42. The database system of claim 29, where further query optimization includes two or more optimizations selected from the group consisting of:

determining a satisfiability of the database query;

determining a transitive closure of the database query;

determining one or more plans for executing the query; and

selecting an optimal plan from executing the database query.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE